



Bluetooth™ in Depth

Session 807





Bluetooth™ in Depth

Michael Larson
Bluetooth Software Manager

Agenda

- Brief description of Bluetooth technology
- Overview of Apple's Bluetooth implementation
- Apple's Bluetooth framework and its objects
- Bluetooth kernel objects



What You'll Learn

- Bluetooth Basics
 - How Bluetooth Works
 - Basic Bluetooth Terminology
- Apple's Bluetooth Implementation
 - Find devices
 - Query services
 - Open L2CAP, RFCOMM, and OBEX connections
 - Extend Apple's kernel implementation to support third-party hardware



Bluetooth Basics

- Bluetooth is a low-bandwidth, short-range wireless protocol
- Bluetooth data transfer happens over a series of layered protocols
 - Baseband
 - L2CAP
 - RFCOMM
 - OBEX
- Devices identified by unique 6 byte addresses—like Ethernet



Bluetooth Basics

Bluetooth Connectivity Modes

- Discoverable
 - Inquiries are used to find other devices within range
 - When discoverable a Bluetooth device will answer to inquiries from other devices
- Connectable
 - Connection requests are used to establish a connection to a remote device
 - When connectable, a remote device will respond to connection requests from other devices
- Connectable and Discoverable are independent states

Bluetooth Basics

Bluetooth Security Concepts

- Pairing relationship
 - A paired device relationship is created by validating a mutually shared secret (passkey)
 - Passkeys are never transmitted over the air
 - Passkeys are used to generate a link key
 - Link Keys validate future connections without user intervention
 - Link Keys are 128-bit values
 - Link Keys can be stored on the computer and/or in the hardware








Bluetooth Basics

Bluetooth Security Concepts

- Encryption
 - A Bluetooth connection can require encryption
 - Encryption is done in the hardware
 - Encryption uses the link key as the seed to the encryption engine
 - 128 bit



Bluetooth Positioning

	Wired	Wireless
Network	 Ethernet	 AirPort
Peripheral	  USB FireWire	 Bluetooth

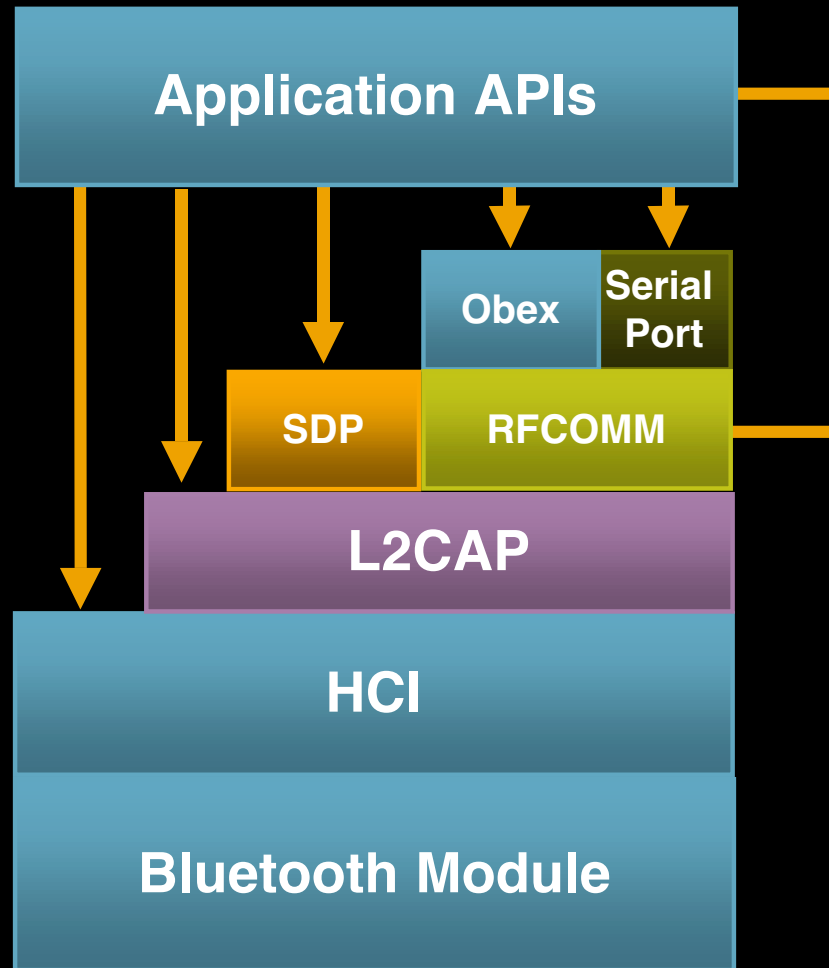


Bluetooth Supported Profiles

- Dial-up Networking (Cell phone client)
 - Internet connect
 - IrDA replacement
- Serial Port
 - Palm OS PDA synchronization
- Object push (OBEX)
 - Enables send and receive of small files
 - vCard push and receive

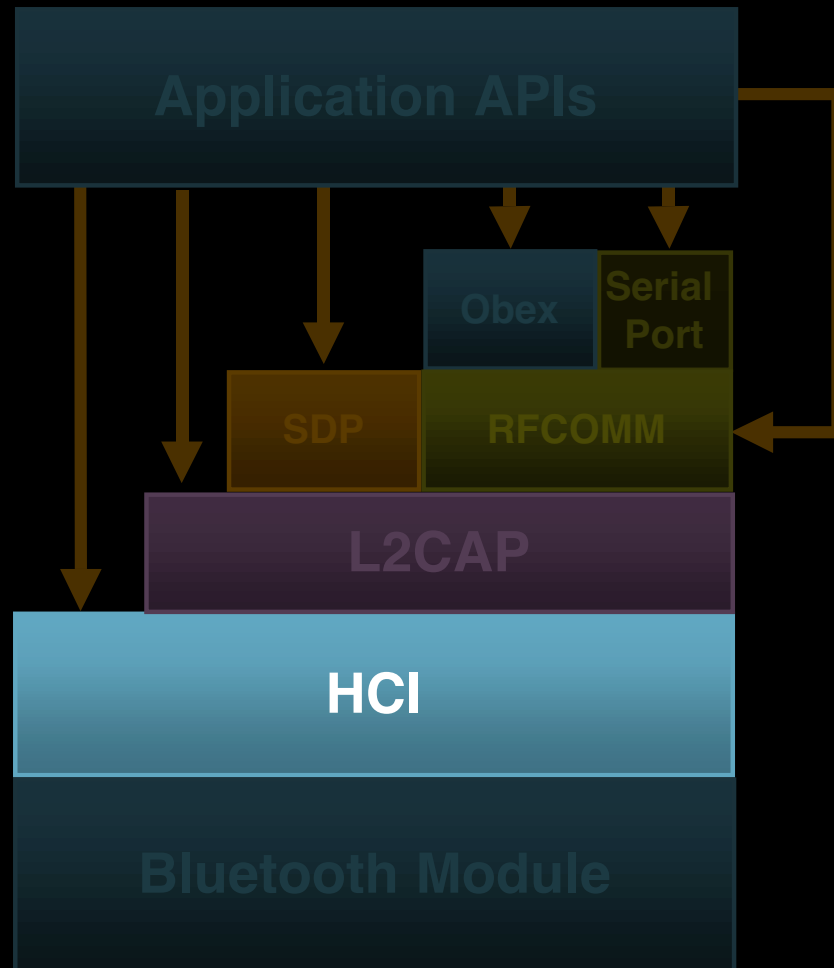


Bluetooth Stack



Bluetooth Basics

HCI—Host Controller Interface

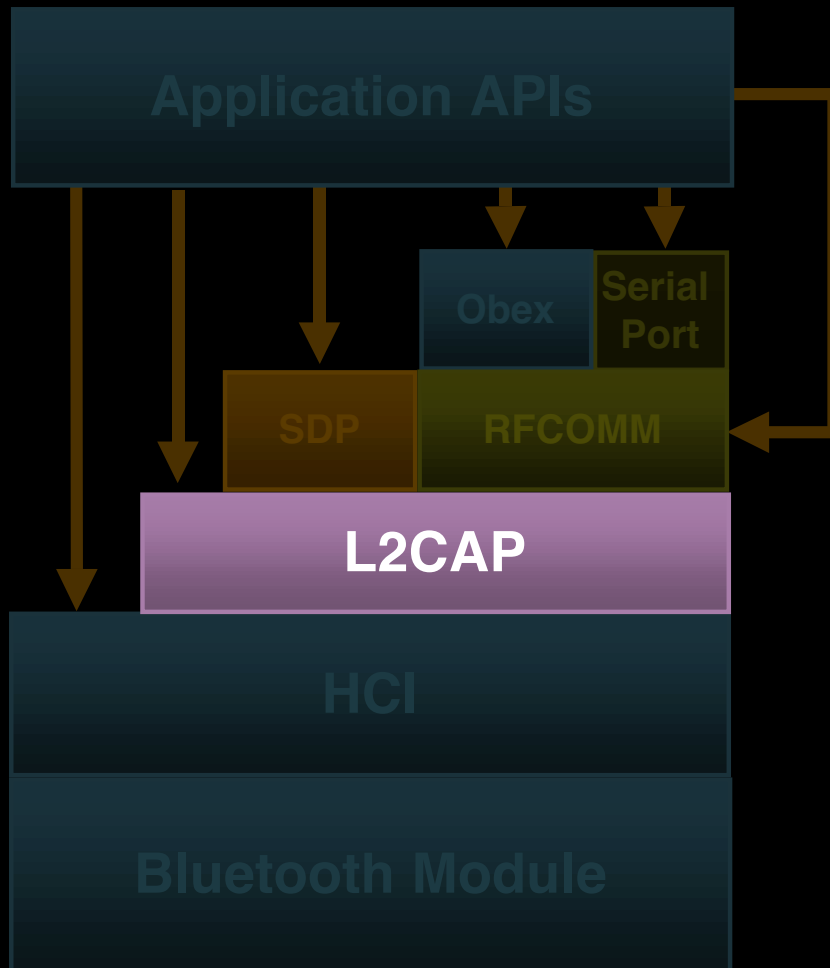


- Abstraction layer to transfer commands, events and data packets to and from the radio module



Bluetooth Basics

L2CAP—Logical Link Control Adaptation Protocol

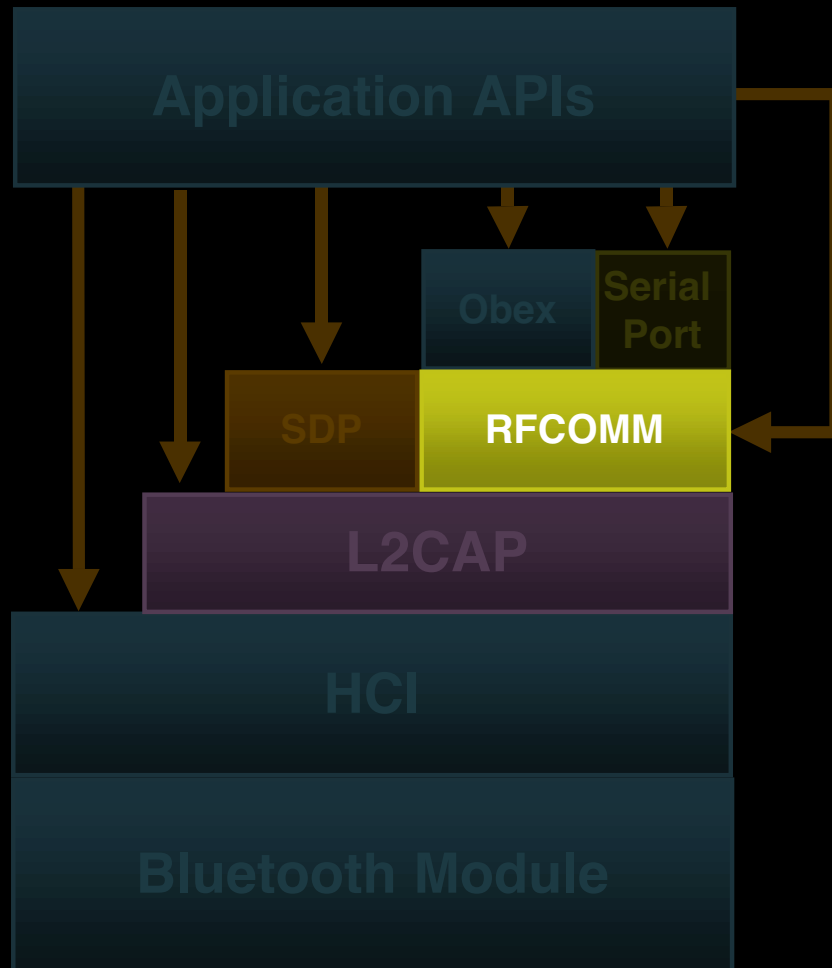


- Simple multiplexed data channel
- Segmentation and re-assembly of data packets



Bluetooth Basics

RFCOMM

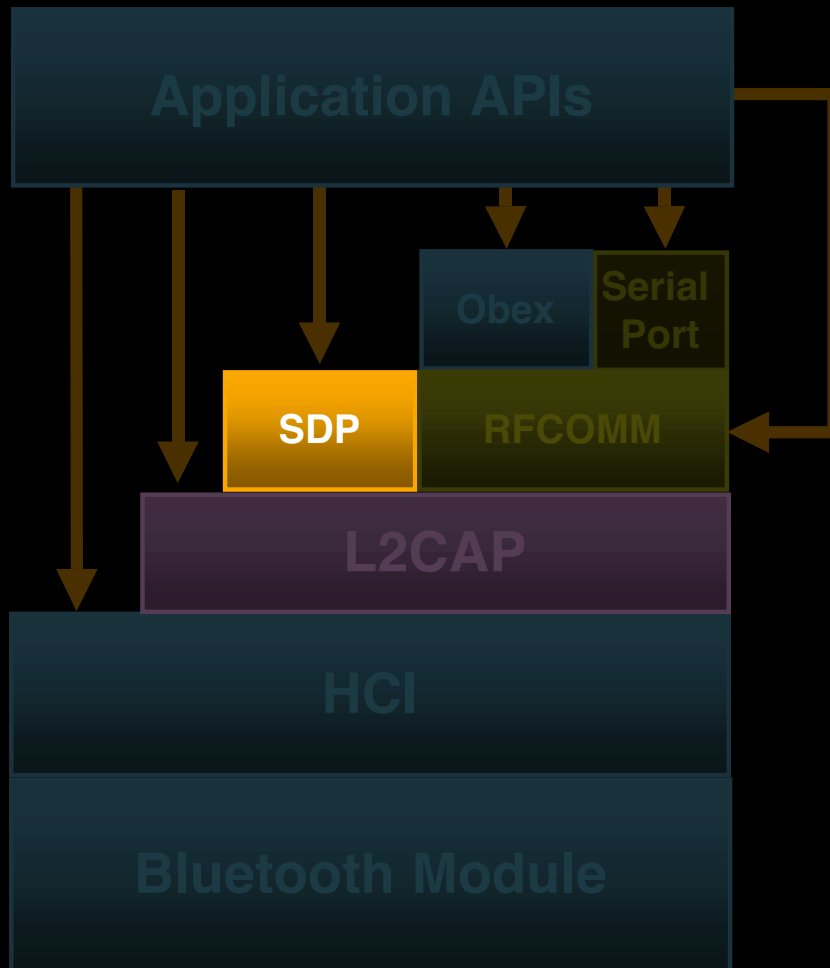


- Serial port emulation layer
- Multiplexed data channel
- Based on ETSI standard 07.10
- Uses a single L2CAP channel as its transport



Bluetooth Basics

SDP – Service Discovery Protocol

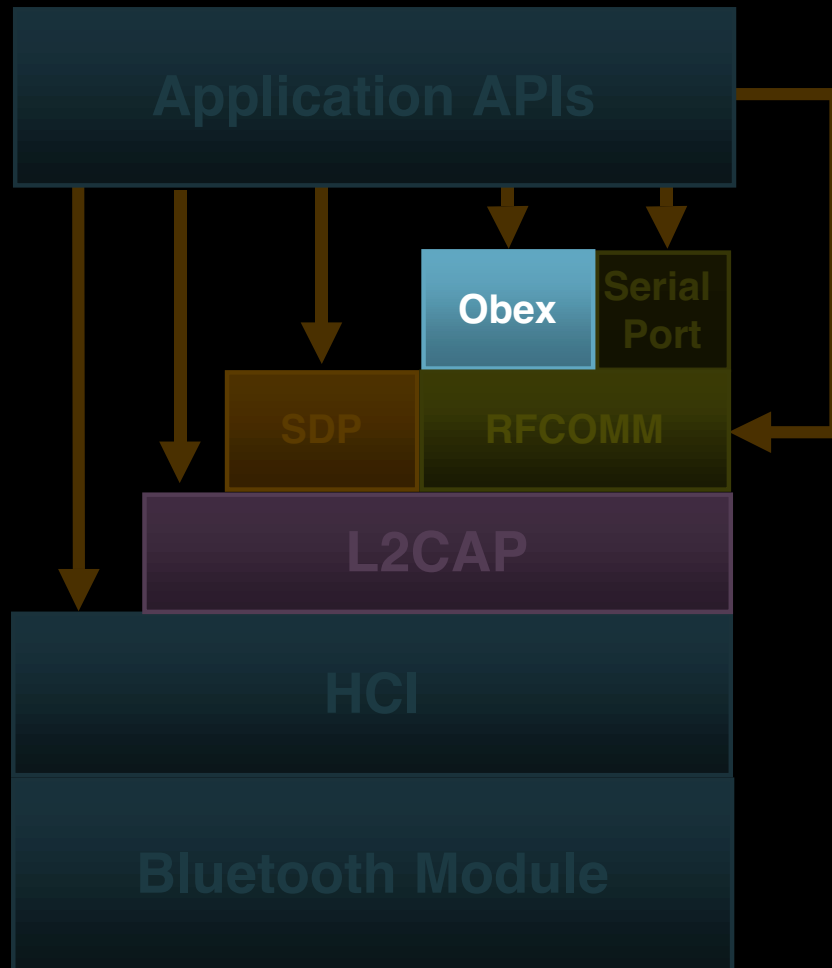


- Protocol for querying a remote device for its supported services and their attributes
- Uses L2CAP channel as transport



Bluetooth Basics

OBEX



- HTTP-like protocol for transferring files or objects
- Based on IrOBEX v1.2 specification
- Uses a RFCOMM Channel as its transport



Apple's Bluetooth SDK

- Bluetooth SDK available online
- Bluetooth PR2 and SDKs posted this afternoon
- Separate Mac OS 10.1.4 and Jaguar Developer release versions
 - Use the correct one
- SDK is a superset of Bluetooth PR2
 - No need to install both



Apple's Bluetooth APIs

- Kernel APIs
 - HCI Controller
 - USB HCI Controller
- User APIs
 - IOBluetoothDevice
 - IOBluetoothL2CAPChannel
 - IOBluetoothRFCOMMChannel
 - OBEXSession
 - Service Discovery Objects



Bluetooth Kernel Objects

- IOBluetoothHCIController
 - Base class for the HCI Controller implementation
 - Subclass this to add support for a Bluetooth radio over a different transport (serial, PCI)
- AppleBluetoothUSBHCIController
 - USB class driver for Bluetooth
 - Can be subclassed to add vendor-specific functionality
- Apple has done USB, other transports are open



Bluetooth User Space Objects

- Object oriented
- C and Objective-C APIs for Bluetooth stack
- Common UI elements to search for devices, and select a service on a device
- Obj-C objects and C objects have similar names
 - Obj-C: **IOBluetoothDevice**
 - C: **IOBluetoothDeviceRef**



IOBluetoothDevice

- Object representing a remote device
 - Baseband Connections
 - L2CAP Channel Connections
 - RFCOMM Channel Connections
 - OBEX Session Connections
 - Service Searches
- Can be returned from UI objects
- Can be created from code from a device address
- Can exist without an open connection



IOBluetoothL2CAPChannel

- Data conduit to a remote device
- Represents an open channel
- APIs to open, read, write and close a channel
- C version: **IOBluetoothL2CAPChannelRef**



IOBluetoothRFCOMMChannel

- API to open, close, read and write a channel
- Event notifications
- Serial emulation layers
 - Supports speed settings for talking to true serial devices
- C version: **IOBluetoothRFCOMMChannelRef**



Service Discovery Protocol

- Creating services for vending
- Getting services from remote device
- SDP-specific classes
 - **IOBluetoothSDPServiceRecord**
 - **IOBluetoothSDPServiceAttribute**
 - **IOBluetoothSDPDataElement**
 - **IOBluetoothSDPUUID**
 - C versions available



OBEX—Object Exchange

- Client/Server session objects via RFCOMM channel
- Current API requires knowledge of OBEX specifications—convenience API coming
- Simple file transfer services are coming
- OBEX Header construction and parsing utilities
- vObject creation utilities
 - vCard for now, others coming soon



Searching for Devices . . .

- Three UI elements
 - Pairing Panel
 - Service Search Panel
 - Device Search Panel
- Common aspects of Bluetooth Panels
 - Ability to specify filters based on device types and service types



Pairing Panel



- Common UI element available to any Bluetooth-aware application
- Searches for devices, and filters according to device type
- Allows a user to create a paired device relationship to any other device



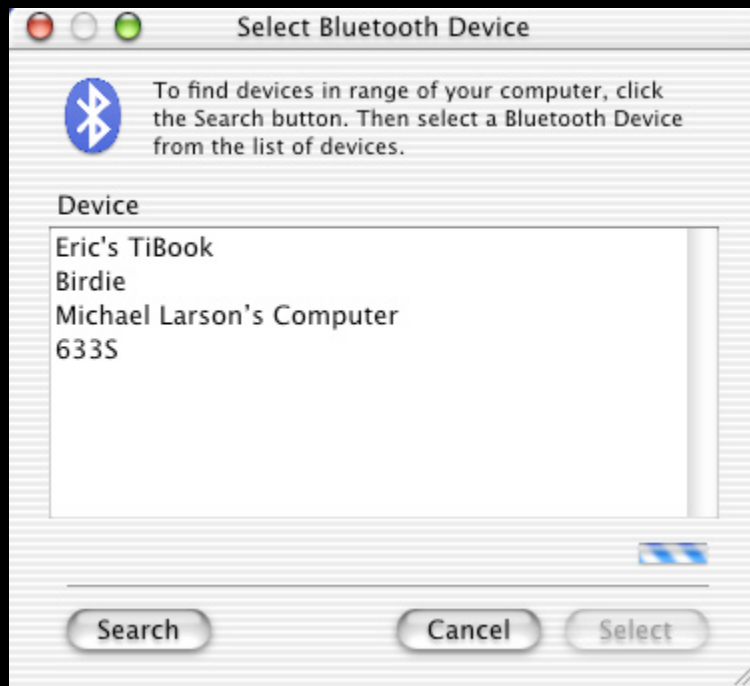
Service Search Panel



- Ability to filter on device type and service type
- Allows the user to select a particular service on a device
- Handles all issues of searching and Service Discovery for the developer



Device Search Panel



- Allows the user to select a device, or multiple devices to use
- Filters on device type, and will optionally validate service availability
- Handles searching and service discovery for the application



C Code Example

```
IOBluetoothDeviceRef          remoteDevice;  
IOBluetoothSDPServiceRecordRef serviceRecord;  
BluetoothRFCOMMChannelID    rfcommChannelID;  
IOBluetoothRFCOMMChannelRef rfcommChannel;  
  
// Bring up the Service Browser UI for the user to select a service...  
IOBluetoothServiceBrowserControllerBrowseDevices(&serviceRecord,  
    kIOBluetoothServiceBrowserControllerOptionsNone);  
  
// Ask the service record for the device  
remoteDevice = IOBluetoothSDPServiceRecordGetDevice(serviceRecord);  
  
// Ask the service record for the RFCOMM channel ID  
result = IOBluetoothSDPServiceRecordGetRFCOMMChannelID(serviceRecord,  
    &rfcommChannelID);
```



C Code Example (Cont.)

```
// Open the baseband connection to the device...
```

```
IOBluetoothDeviceOpenConnection(remoteDevice);
```

```
// Open the RFCOMM channel
```

```
IOBluetoothDeviceOpenRFCOMMChannel(remoteDevice ,rfcommChannelID,  
    &rfcommChannel);
```

```
// Register for incoming RFCOMM events (including incoming data).
```

```
IOBluetoothRFCOMMChannelRegisterIncomingEventListener(rfcommChannel,  
    rfcommEventListener, myRefCon);
```

```
// write some data to the RFCOMM channel...
```

```
IOBluetoothRFCOMMChannelWrite(rfcommChannel, buffer, length, TRUE);
```

```
// Declaration of the RFCOMM event listener...
```

```
void rfcommEventListener (IOBluetoothRFCOMMChannelRef rfcommChannel,  
    void *refCon, IOBluetoothRFCOMMChannelEvent *event);
```



Obj-C Code Example

```
IOBluetoothDevice                *remoteDevice;  
IOBluetoothSDPServiceRecord    *serviceRecord;  
BluetoothL2CAPPSM              l2capPSM;  
IOBluetoothL2CAPChannel        *l2capChannel;  
  
// Bring up the Service Browser UI for the user to select a service...  
[IOBluetoothServiceBrowserController browseDevices:&serviceRecord  
  options:kIOBluetoothServiceBrowserControllerOptionsNone];  
  
// Ask the service record for the device  
remoteDevice = [serviceRecord getDevice];  
  
// Ask the service record for the L2CAP PSM  
[serviceRecord getL2CAPPSM:&l2capPSM];
```



Obj-C Code Example (Cont.)

```
// Open the L2CAP channel - this will open the baseband connection  
[remoteDevice openL2CAPChannel:l2capPSM findExisting:FALSE  
              newChannel:&l2capChannel];  
  
// Register for incoming L2CAP data  
[l2capChannel registerIncomingDataListener:myDataListener refCon:myRefCon];  
  
// write some data to the L2CAP channel...  
[l2capChannel write:buffer length:length];
```



Sample Apps in SDK

- OBEXQuickPush
- OBEXSample
- OBEXSampleSendVCard
- RFCOMMClientSample
- RFCOMMServerSample



Tools Available in SDK

- Packet Decoder2
 - Logs all HCI commands and events
 - Logs all ACL data packets
 - Decodes ACL packets as L2CAP, RFCOMM or SDP
 - Possibility for other decoding options later
- Bluetooth Monitor
 - Lists all open connections and channels
 - Ability to force close a channel or connection



Roadmap

806 Wireless Directions:

Future directions for Apple's wireless products

Room A1

Wed., 9:00am



Who to Contact

Thomas Weyer

Network and Communications Evangelist

weyer@apple.com

Bluetooth Developer Mail List

bluetooth@lists.apple.com

<http://www.lists.apple.com/bluetooth>

<http://developer.apple.com/wwdc2002/urls.html>



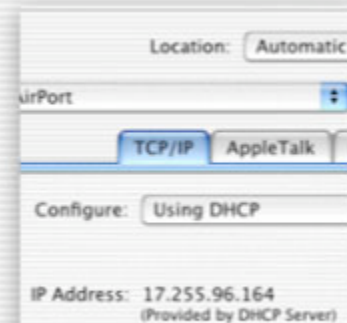
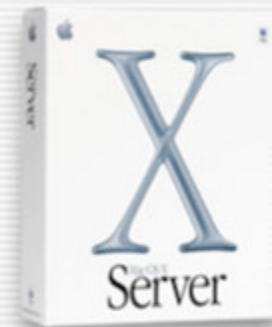
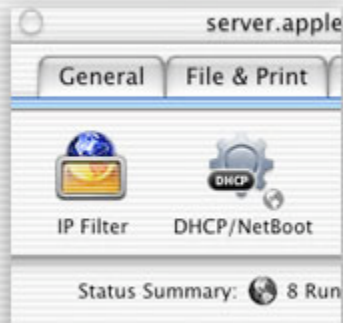
For More Information

- Apple's Bluetooth SDK
<http://developer.apple.com/sdk>
- Bluetooth Specification Version 1.1
<http://www.bluetooth.org>
- GSM 07.10 Specification for RFCOMM v6.3.0
<http://www.etsi.org>
- OBEX Specification
<http://www.irda.org/standards/specifications.asp>
- "Bluetooth Connect Without Cables,"
 - Bray, Sturman, Mendolia • ISBN: 0130661066





Q&A



Tom Weyer
Network and Communications Evangelist
weyer@apple.com

<http://developer.apple.com/wwdc2002/urls.html>

 **WWDC2002**

 **WWDC2002**

 WWDC 2002